



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/520,405	03/08/2000	Michael G. Martinek	307.029US1/PA0390	1300

7590

08/02/2002

Schwegman Lundberg Woessner & Kluth PA
P O Box 2938
Minneapolis, MN 55402

EXAMINER

ASHBURN, STEVEN L

ART UNIT

PAPER NUMBER

3714

DATE MAILED: 08/02/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/520,405

Applicant(s)

MARTINEK ET AL.

Examiner

Steven Ashburn

Art Unit

3714

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 13 May 2002.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-54 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-54 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 08 March 2002 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____
- 4) ☐ Interview Summary (PTO-413) Paper No(s) _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

MARK SAGER
PRIMARY EXAMINER

DETAILED ACTION

Drawings

The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the following features must be shown or the feature(s) canceled from the claims. No new matter should be entered.

- a. System handler application. *See e.g. claim 1.*
- b. Dynamically linking a gaming program object. *See e.g. claim 1.*
- c. Loading a gaming program shared object and executing the gaming program object. *See e.g. claim 3.*
- d. Storing game program objects in non-volatile memory. *See e.g. claim 4.*
- e. Executing a callback function corresponding to a change of game data in non-volatile memory. *See e.g. claim 5.*
- f. Loading a shared object, executing a shared object, and accessing and storing game data in nonvolatile memory, unloading a shared object and loading another second shared object. *See e.g. claims 13, 21.*
- g. System handler and kernel working in communication to hash the system handler code and operating system kernel code. *See e.g. claim 38.*
- h. Control of a networked, online system. *See e.g. claim 45.*
- i. Control of a progressive meter. *See e.g. claim 46.*
- j. ROM. *See e.g. claim 47.*
- k. Accessing user level code from ROM, executing user level code from ROM, zeroing out unused RAM, testing and/or hashing the kernel, and disabling selected device handlers. *See e.g. claim 47.*

Art Unit: 3714

- I. Dynamically linking an Application program interface to the system handler. *See e.g. claim 52.*

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 103

Claims 1-4, 6, 10-13, 16, 18-21, 23-28, 30-33, 35, 38-43, 48, and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell*, U.S. Patent 6,075,939 (Jun. 13, 2000).

Bunnell discloses a UNIX-based operating system with a scalable kernel that is customizable to support specialized applications. *See fig. 1, 2, 2:66-3:3.* The system's modularity allows support it to support a range of applications ranging from small, embedded-type systems to large, network-capable instantiations. *See col. 2:66-3:4.* Furthermore, the modularity allows developers to implement special purpose kernel components in support of specialized applications and devices. *See col. 3:24-4:2.* Accordingly, the size of the operating system's kernel may be scaled to support various applications. *See col. 8:37-63.* Thus, a developer may incorporate whatever optional kernel components desired, including incorporating application programs to maximize the system's resources by reducing computational overhead. *See id.* *Bunnell* describes the following features of the claimed invention:

- a) System handler, executed by the operating system kernel, operable to dynamically link with at least on a program object. *See fig. 1; 3:4-10; and 3:43-58. (Claim 1, 39, 48)*
- b) Operating system comprises a system handler. *See figs. 1-5. (Claim 30)*
- c) System handler comprising a plurality of device handlers. *See figs 1-5. (Claim 2, 31, 42)*
- d) System handler loads and executes program objects. *See fig. 1-5; See 3:43-59; 9:50-54; 11:20-12:15. (Claims 3, 13, 21)*
- a) Kernel modifications are modular. *See 3:24-30. (Claim 10, 38)*

Art Unit: 3714

- e) A plurality of shared objects and a system handler adapted to execute at least one shared object called from memory. *See fig. 1-4; col. 5:25-46 (Claim 19)*
- f) Shared objects describing a personality in a selected mode. *See col. 3:31-43, 4:47-57, 12:1-3. (Claim 28)*
- g) System handler comprises APIs with a library of functions callable from program objects. *See fig. 4; col. 3:43-59, 9:50-54, 11:20-12:15. (Claims 11, 26, 33, 35, 40)*
- h) System handler comprises manages an event queue determining the order of the device handlers. *See fig. 1, 2; 3:39-42; 5:26-46, 6:26-47. (Claims 12, 23, 30, 32, 34, 41)*
- i) Operating system kernel executing on a computerized controller comprising an element of a universal operating system also comprising a system handler. *See fig. 1, 2. (Claim 48)*

Bunnell does not teach the claimed subject matter of:

- a) Game controller operable to control a wagering game *(Claims 1, 16, 43)*
- b) Gaming programs as shared objects. *(Claim 3)*
- c) Storing game data in non-volatile storage. *(Claims 4, 18, 20, 27, 49)*
- d) An IBM PC-compatible controller. *(Claims 6,)*
- e) Operating system kernel customized for gaming use *(claim 16)*
- f) Executing only one program object at a time. *(Claims 24, 25).*
- g) Event queue queuing on first-come, first-server basis. *(Claim 36)*
- h) Event queue queuing using more than one criteria. *(Claim 37)*

Regardless of the deficiencies, these features would have been obvious to a gaming artisan.

In regards to claims 1, 16, 43, 48, *Bunnell* suggests a scalable operating system supporting specialized computing applications that flexibly incorporates operating system modules into the kernel to provide an efficient and stable structure for customized applications such as embedded systems, custom

Art Unit: 3714

hardware and OEM devices. *See fig. 4; col. 2:59-3:3:3.* For example, *Bunnell* suggests customizing the kernel for embedded systems such as point-of-sale or security systems. *See col. 8:15-19.*

A gaming artisan at the time of the invention would possess knowledge of various operating systems, and an understanding of their fundamental features and relative advantages. Thus, it would have been obvious to a gaming artisan at the time of the invention to modify the customizable operating system disclosed by *Bunnell's* for use in gaming devices to provide a cheap, flexible, efficient and stable system and thereby enhance the device's performance, reliability and flexibility.

In regards to claims 3, 16, 28, 40 and 49, *Bunnell* allows developers to implement special purpose kernel component in support of custom applications and devices. *See col. 3:24-4:2.* Accordingly, the system handler loads and executes program objects required for an application as defined in a profile. *See fig. 1-5; col. 4:47-58.*

The Applicants' specification defines "gaming program shared objects" as self contained functional units of game code that define the a particular feature or sequence of operation for a game wherein the personality and behavior of a gaming machine are defined by a set of shared objects. *See p. 7, ¶ 4.* These shared objects are equivalent to the modular operating system components described by *Bunnell* as an application profile.

Furthermore, it notoriously well known to employ Object Oriented Programming to structure computer programs into independent function modules to reduce the time and expense required to develop and modify computer software by reusing object module.

Thus, it would have been obvious to a gaming artisan at the time of the invention to modify *Bunnell*, wherein a specialized operating system executes an operating system according to a customized profile, to load and execute gaming program shared objects to provide an efficient and stable operating

Art Unit: 3714

system customized to the special purpose of gaming and thereby reduce the time of expense of developing game software by reusing game object modules.

In regards to claims 4, 13, 18, 20, 27 and 49 it is notoriously well known in the art of gaming to store and access game data in non-volatile memory to prevent loss and/or modification of data in the event of a power loss or other fault. The feature enhances the robustness and security of a gaming machine. First, it avoids conflict between players and operators over the state of the game prior to power loss or fatal fault in the execution of the software. Second, it complicates attempts to fraudulently tamper with the game data. Thus, it would have been obvious to a gaming artisan at the time of the invention to adapt the customizable operating system suggested by *Bunnell* for gaming by adding the feature of storing game data in non-volatile memory and thereby protecting the loss or tampering of game data in the event of power loss or other faults.

In regards to claims 6 and 29, it was notoriously well known at the time of the invention to employ PC-based systems in custom applications because of their cost, availability, flexibility, compatibility and broad commercial support. Furthermore, a gaming artisan at the time of the invention would have possessed knowledge of the various commercially available computer controllers, as well as their fundamental features and relative advantages. Still furthermore, it was well known in the gaming art at the time of the invention to employ personal computers as game controllers. Thus, it would have been obvious to a gaming artisan at the time of the invention to modify the operating system suggested by *Bunnell* to implement gaming systems with IBM PC-compatible controllers to decrease the cost and increase the supportability of a gaming device.

In regards to claims 24 and 25, *Bunnell* discloses that the customizable operating system capable of co-execution of multiple processes. *See col. 3:4-20*. Regardless of the advanced capability offered by *Bunnell*, it is fundamental, typical, and notoriously well known for computers to execute only one program object at a time. Thus, it would have been obvious to one of ordinary skill to implement an application using the operating system discloses by *Bunnell* such that only one program object is executed at a time to increase the simplicity and robustness of application software.

In regards to claims 36 and 37, *Bunnell* discloses an customizable operating system with an event queue, however it does not describes the specifics of the queue management. Thus, *Bunnell* describes all the features of the instant claims except the event queue queuing on first-come, first-server basis (*claim 36*) and the event queue queuing using more than one criteria (*claim 37*). Regardless of the deficiencies, the features were known in the art at the time of the invention and would have been obvious to an artisan.

Event queues are a basic function of an operating system for managing the system's response to events. It is fundamental programming technique to arrange the event response protocol to respond to events in the most effective order. For example, in some cases, event queues are simply organized on a first-serve/first-serve basis. In other cases, the some events are more critical than others. Thus, the event queues are simply organized on a priority basis. In still other cases, events are handled on both a priority and first-come/first-serve basis. It would be a matter of design choice as to which manner the event queue managed events.

Thus, it would have been obvious to an artisan at the time of the invention to modify the gaming operating system suggested by *Bunnell* to manage to event queuing on first-come, first-server basis or using more than one criteria to manage the event queue's priorities to respond to common events, such as button presses, on a first-come/first-serve basis while responding to critical events, such as security faults, immediately based on a higher priority.

Art Unit: 3714

Claims 7, 8, 14, 21 and 52-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell* in view of David A. Rusling, *The Linux Kernel*, <<http://www.tldp.org/LDP/tlk/tlk.htm>> (1999) (hereinafter "*Rusling*").

Bunnell suggests a gaming device using a UNIX-based operating system with a kernel that is modifiable for custom applications. *See col. 13:34-43*. The reference suggests the system is equally applicable to other general purpose and embedded systems. *See id.* Hence, the gaming device disclosed by *Bunnell* suggests all the features of the claimed subject matter except:

- a. Using a LINUX operating system kernel. (*claims 7, 8*),
- b. Unloading first program object, and unloading a second program object (*claim 14*)
- c. Dynamically linking eh application program interface by the system handler. (*Claims 52-54*)
- d. Executing only one program object at a time. (*Claim 25*)

Regardless of the deficiencies, these features would have been obvious to a gaming artisan in view of *Rusling*.

In regards to claims 7 and 8, *Rusling* describes the fundamental features of the LINUX operating system. The system is a well-known, freely distributed, PC-compatible operating system derived from UNIX. Analogously to *Bunnell*, the LINUX kernel is scalable and customizable for specialized applications. A gaming artisan at the time of the invention would possess knowledge of the fundamental features and relative advantages of the various operating systems, including LINUX. Thus, it would have been obvious to gaming artisan at the time of the invention to modify *Bunnell*'s operating system to be based upon a LINUX foundation and thereby improve the gaming device by using an operating system customized for gaming using a scalable, free and widely supported operating system.

In regards to claims 14 and 52-54, *Bunnell* suggests that application-level program objects may be incorporated within the operating system's kernel. *See col. 8:37-63*. The reference suggests that it is a matter of design choice to modularly link various levels of the system software with the kernel depending on the accessibility required of the user.

Rusling describes the LINUX operating system's capability to dynamically load and unload shared kernel modules as needed. *See chapt. 12, ¶¶ 1-13*. Dynamically loading the modules advantageously minimizes the size of the kernel, allows flexible execution and thereby reduces the demand for system resource. *See id.* Hence, the operating system is tailored to dynamically link modules into the operating system to minimize the resource required by the system.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the gaming system suggested by *Bunnell*, wherein the system may be customized to link various levels of code into the kernel, to add the feature of dynamically loading and unloading of program object modules, including API objects, into the kernel in order to provided an efficient and flexible gaming system wherein users require little access to the application code.

Claims 5 and 15, are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell* in view of *Pascal*, U.S. Patent 5,971,851 (Oct. 26, 1999).

Claims 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell* in view of *Rusling*, as applied to claims 7, 8, 13 and 22 above, in further view *Pascal*.

The gaming device suggested by either *Bunnell* or the combination of *Bunnell* with *Rusling* describes all the features of the claimed subject matter except executing a callback function corresponding to a change in game data stored in non-volatile memory. Regardless of the deficiency, the feature would have been obvious to a gaming artisan in view of *Pascal*.

Art Unit: 3714

Bunnell additionally describes a kernel module providing security management for the system. *See col. 16-29*. The module monitors applications for access control and reporting. *See id.* Hence, *Bunnell* generally suggests monitoring application modules to ensure the security of the systems.

Pascal discloses an analogous operating system for a gaming device wherein callbacks are employed to communicate information between application modules upon the occurrence of certain events. *See 1:44-2:30*. In generally, callback routines are used in state-based machines to communicate data between independent modules upon the occurrence of predetermined events. *See col. 6:25-45*. *Pascal* describes using callback to enhance the robustness of a gaming device under fault conditions to protect data that may affect the outcome of a game payout. *See col. 2:25-30*.

In view of *Pascal*, it would have been obvious to an artisan at the time of the invention to modify the customized gaming operating system suggested by either *Bunnell* or the combination of *Bunnell* with *Rusling*, wherein the system enhance security by monitoring application modules, to execute a callback function corresponding to a change in game data stored in non-volatile memory to enhance the security of the gaming device by monitoring changes in data that might affect the outcome of the game payout and thereby provide a more secure gaming device that is resistant to errors caused by losses in power or tampering.

Claims 9, 10, 17, 38, 44 and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell* in view of *Pascal*, as applied to the claims 22, 23 above, in further view of *Bock*, U.S. Patent 5,155,856 (Oct. 13, 1992) and *Davis*, U.S. Patent 6,401,208 (Jun. 4, 2002).

Bunnell discloses a scalable, modular operating system that selectively initializes only the parts of the kernel required under a system profile and does not attempt to initialize components that are not present. *See col. 11:20-28, 12:17-23, 13:3-12*. As a result, the system advantageously allows the

Art Unit: 3714

operating system to configure kernel components in or out of the system. *See id.* Hence, *Bunnell* describes a system that disables selected device handlers.

Bunnell additionally discloses that the operating system can incorporate user-level code on several levels. For example, in a fully stand-alone embodiment, where only a single user-level application is executed, the application may be coupled directly with the operating system to allocate all the system resources to executing the single application. *See col. 8:39-46.* Accordingly, *Bunnell* does not limit what applications the developer includes within the kernel. *See 8:47-56.* Hence, *Bunnell* suggest coupling a user-level application with the operating system that is executed upon bootstrapping the system.

Pascal describes the process of executing an operating system from ROM upon bootstrapping a gaming device from a cold or warm start. *See fig. 2; col. 3:41-67.*

Thus, the operating system for a gaming device suggested by the combination of *Bunnell* with *Pascal* describes all the features of the present claims except zeroing-out unused RAM and testing and/or hashing the kernel. Regardless of the deficiencies, these features were known in the art at the time of the invention and would have been obvious to an artisan in view of *Bock* and *Davis*.

Bock describes “self-guarding” computer system that, upon initialization, zeros-out unused memory to erase information which is not required for subsequent operation. *See col. 1:38-62.* Erasing the data serves to eliminate vestigial data remaining in memory after reset that may interfere with the newly executed instructions and cause errors. *See col. 3:10-20.* Hence *Bock* suggests enhancing the security of a computing system by zeroing-out unused memory.

Davis describes a computing system that, upon initialization, tests and/or hashes the BIOS to enhance to system’s security. The reference describes several threats to computing device that circumvent security measures implemented at higher levels of operation. For example, a malefactor may

Art Unit: 3714

replace the device's ROM or a software virus may infiltrate the BIOS. In solution of this threat, *Davis* suggests testing the BIOS using a hash-function upon initialization. *See col. 1:32-2:5.*

The above references analogously describe means for initializing and increasing the security of an operating system. In view of *Bock* and *Davis*, it would have been obvious to a gaming artisan at the time of the invention to modify the gaming device operating system suggested by the combination of *Bunnell* with *Pascal* to add the features of zeroing-out unused RAM and testing and/or hashing the kernel. The modification would enhance the security of the gaming device by frustrating attempts to tamper with the operating system by modifying the data stored within the device's memory prior to initialization.

Claims 45, 46, 50 and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Bunnell* in view of *Wiltshire*, U.S. Patent 6,409,602 (Jun. 25, 2002).

Bunnell teaches the operating system is scalable to be network compliant. *See col. 2:27-3:3.* As describes above, the reference describes all the features of the claimed subject matter except using the operating system to control a networked online system (*claim 45*) and using the system to control a progressive meter (*claim 46*). Regardless of the deficiencies, the features were known in the art at the time of the invention and would have been obvious to an artisan in view of *Wiltshire*.

Wiltshire discloses an analogous gaming system in which PC-based machines execute commercially available operating systems to decrease the cost of developing and upgrading gaming devices. *See col. 2:6-44, 4:44-7:5.* The reference describes an operating system controlling a networked, online gaming system including managing a progressive jackpot. *See col. 4:66-5:13, 5:45-64.*

In view of *Wiltshire*, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the gaming operating system suggested by *Bunnell*, wherein the operating system is scalable to support networks, to add the features of controlling a networked online system and

Art Unit: 3714

controlling a progressive meter. The modification would allow the gaming device to communicate over a casino networks and thereby make the device attractive to operators whose casinos link gaming machines to networks for player tracking, accounting, monitoring, and linking to progressive jackpots.

Response to Arguments

Applicant's arguments with respect to claims 1-55 have been considered but are moot in view of the new grounds of rejection. Regardless, the examiner responds to the Applicants' general arguments below.

In response to the Applicants' argument that *Bunnell* is non-analogous art because it falls outside the field of gaming, the Examiner respectfully disagrees. It has been held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the applicant was concerned, in order to be relied upon as a basis for rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In this case, the applicant claims an operating system with a kernel customized to support gaming devices. In comparison, *Bunnell* describes an operating system with a kernel customizable to support specialized applications. See *fig. 1, 2; col. 2:59-3:3*. *Bunnell* further describes potential uses for the operating system in embedded systems such as point-of-sale and security systems. See *col. 8:15-19*. Broadly viewed, point-of-sale and security systems employ features analogous to gaming devices. In particular, gaming devices typically employ embedded systems of the type suggested by *Bunnell*. Thus, *Bunnell* is analogous art because it is reasonably pertinent to the particular problem with which the applicant was concerned.

Art Unit: 3714

In response to the Applicants' argument that it would not be obvious to a gaming artisan to modify a personal computer for gaming purposes, the examiner respectfully disagrees. The standard of patentability is what the prior art taken as a whole at a time prior to the invention suggests to an artisan. In this case, *Bunnell* describes an operating system with a kernel customizable to support specialized applications. See *fig. 1, 2; col. 2:59-3:3*. *Bunnell* further describes potential uses for the operating system in similar systems such as point-of-sale and security systems. See *col. 8:15-19*.

It was well known at the time of the invention to employ PC-based systems in custom applications because of their cost, availability, flexibility, compatibility and broad commercial support. Furthermore, a gaming artisan at the time of the invention would have possessed knowledge of the various commercially available computer controllers, as well as their fundamental features and relative advantages. Still furthermore, it was well known in the gaming art at the time of the invention to employ personal computers as game controllers. See, e.g. *Wiltshire, col. 4:49-5:28*.

Gaming developers are required by to meet gaming regulations enacted to ensure gaming devices provide critical security functions protecting internal and external communications. Thus, a gaming artisan is highly motivated to incorporate these features in a gaming device in order to attain regulatory approval.

Thus, when the teachings of *Bunnell*, gaming regulations, and prior PC-based gaming systems are taken as whole, the prior art suggests to a gaming artisan at a time prior to the invention modifying a personal computer for gaming purposes.

Conclusion

The following prior art is considered pertinent to applicant's disclosure of record, but not relied upon:

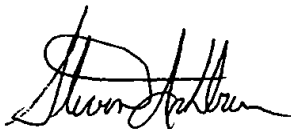
Art Unit: 3714

Fullerton, U.S. 4,607,844 (Aug. 26, 1986) describes a gaming device storing game data in non-volatile memory.

Object-Oriented Programming Concepts, Sun Microsystems, Inc (2002) describes fundamentals of employing program objects in software applications.

Terry Monlick, *What is Object-Oriented Software*, Software Design Consultants, LLC. (1999) describes fundamentals of employing program objects in software applications.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Steven Ashburn whose telephone number is 703 305 3543. The examiner can normally be reached on Monday thru Friday, 8:00 AM to 4:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Valencia Martin-Wallace can be reached on 703 308 4119. The fax phone numbers for the organization where this application or proceeding is assigned are 703 872 9302 for regular communications and 703 872 9303 for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703 308 1078.



Steven Ashburn
July 16, 2002



MARK SAGER
PRIMARY EXAMINER